Lynch, Stephen ORCID logoORCID: https://orcid.org/0000-0002-4183-5122 (2020) Programming in the Mathematics Curriculum at Manchester Metropolitan University. MSOR Connections, 18 (2). pp. 5-12. ISSN 1473-4869

Downloaded from: https://e-space.mmu.ac.uk/626138/

Version: Published Version

Publisher: Educational Development Unit, University of Greenwich

DOI: https://doi.org/10.21100/msor.v18i2.1105

Please cite the published version

https://e-space.mmu.ac.uk

CASE STUDY

Programming in the Mathematics Curriculum at Manchester Metropolitan University

Stephen Lynch, Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK. Email: s.lynch@mmu.ac.uk. https://orcid.org/0000-0002-4183-5122.

Abstract

An increasing number of schools are teaching programming to their pupils and there is also an increase in programming in Higher Education with recent reports recommending this approach. At Manchester Metropolitan University (MMU) we wanted to attract and retain mathematics students and prepare them for careers upon graduation. By integrating Mathematics/Statistics/Operational Research packages across the curriculum and by solving real world problems we have managed to make the course highly desirable and loved by our students. In this case study, we show how it is possible to integrate programming and mathematical/computational modelling across the curriculum.

Keywords: Coding, computational modelling, real-world problems.

1. Introduction

In the 1990s, the mathematics degrees at MMU were technology based due to its historical polytechnic status. As technology became more and more advanced the staff found that they needed this technology to solve real-world problems. At that time, staff were using Maple™, MATLAB® and Mathematica® for mathematics, Minitab® for statistics and, for operational research, staff had developed their own software. There was not much programming involved in the course and students were being shown how technology could help solve problems in isolated pockets of the curriculum. After Curriculum 2000 was introduced into the UK, numbers on mathematics degrees across the country, including at MMU, dipped alarmingly and that is when we at MMU decided that a radical new approach was needed to draw students onto our mathematics degrees. It was also at this time that MMU decided to save money and pay for only one software licence for the faculty. Engineering has always been a much larger department than mathematics at MMU, and so it was decided that the university would adopt MATLAB as its sole programming language for all departments in the Faculty of Science and Engineering. In our new curriculum in 2000, we adopted MATLAB in mathematics and established a new first year unit entitled “Linear Algebra and Programming Skills.” The other mathematics units were also adapted to incorporate MATLAB, the statistics units continued with Minitab, and as operational research staff retired, some MATLAB and open-source software was adopted. As the curriculum continues to develop, we have found that programming is playing an increasingly important role within our courses.

Initially, there was some resistance from some of the students with regards to programming and the majority of students chose units without much coding, however, as staff became more experienced at teaching programming, there was a slow migration over to those units and now programming is popular on the mathematics degrees. We were delighted to read the government reports published in 2018 which vindicated our approach to teaching.

In February 2018, the Government Office for Science and the Council for Science Technology published the Blackett Review (Peplow, 2018), a report highlighting the importance of simulation utilizing computational and mathematical modelling of complex systems in both the public and private sectors. A few months later, in April 2018, an independent review of knowledge exchange in the mathematical sciences was published entitled "*The Era of Mathematics - Review Findings on Knowledge Exchange in the Mathematical Sciences"* (Bond, 2018). Amongst the findings and recommendations of the Bond review was the recommendation that – "*All mathematics students should acquire a working knowledge of at least one programming language*." Both of these reports highlight the importance of computational modelling and programming in the modern world. With mathematics packages such as Maple, MATLAB, Mathematica and Python; statistics packages such as Minitab, R, SAS, SPSS; and operational research packages such as CPLEX, FICO Xpress and Gurobi, mathematicians have the tools necessary to respond to these needs. With regards to computational modelling, the reader may be interested in the following simulation packages MapleSim™, Simulink® and Wolfram SystemModeler® developed by Maplesoft™, MathWorks® and Wolfram®, respectively. Python does have a framework for modelling and simulating dynamical systems called SimuPy, but this has only recently been developed. For big data, machine learning and deep learning, the reader is directed to PyTorch and TensorFlow, and finally, for operational research simulation, Simul8 currently appears to be the most popular tool.

At Visit Days and Open Days at MMU we are finding an increasing number of students have experience at programming. Scratch is used in many primary and secondary schools around the UK. As of October 2019, the community statistics on the Scratch web pages show over 45,000,000 projects shared with more than 47,000,000 users registered (Scratch MIT, 2019). The Raspberry Pi is used in many secondary schools around the UK. It promotes both Python and Scratch as the main programming languages and there is also support for other languages. The official magazine for the Raspberry Pi is the MagPi magazine and there are regular Raspberry Jam events held in the UK and across the world. Amongst the many projects available to school children are: Replace your desktop PC with a Raspberry Pi; Print with Your Raspberry Pi; Set Up a Retro Gaming Machine; Build a Minecraft Game Server; Control a Robot; Broadcast a Pirate FM Radio Station; Build a Raspberry Pi Web Server and Learn How to Code. See (Balon, 2019) for more examples of the Raspberry Pi in education.

Programming in mathematics in Higher Education is becoming increasingly important in this technological age. In 2017, Sangwin and O'Toole (2017) conducted an online survey, with significant follow up correspondence, to establish the number of mathematics courses within the UK which incorporate programming on their degree program. The survey was completed by 63% of those institutons which offer a BSc (Hons) Mathematics degree. It was found that 78% of respondents included programming in a compulsory module and 11% did not teach programming at all. There are some universities in the USA which integrate programming into the undergraduate mathematics courses, see for example, (Buteau et al., 2015), (Cline et al., 2019) and (Jones et al., 2019).

2. Curriculum Design

At MMU, we have incorporated programming and computational modelling into the curriculum for more than twenty years. Figure 2 shows a typical four-year MMU MMath options map for the curriculum. The boxes coloured blue are heavily mathematics package-based, and packages such as Maple, Mathematica, MATLAB and Python would all serve equally well. The boxes coloured green are statistics based and boxes coloured yellow are operational research based. Suitable packages are again indicated in the legends in the top right-hand corner of Figure 1.

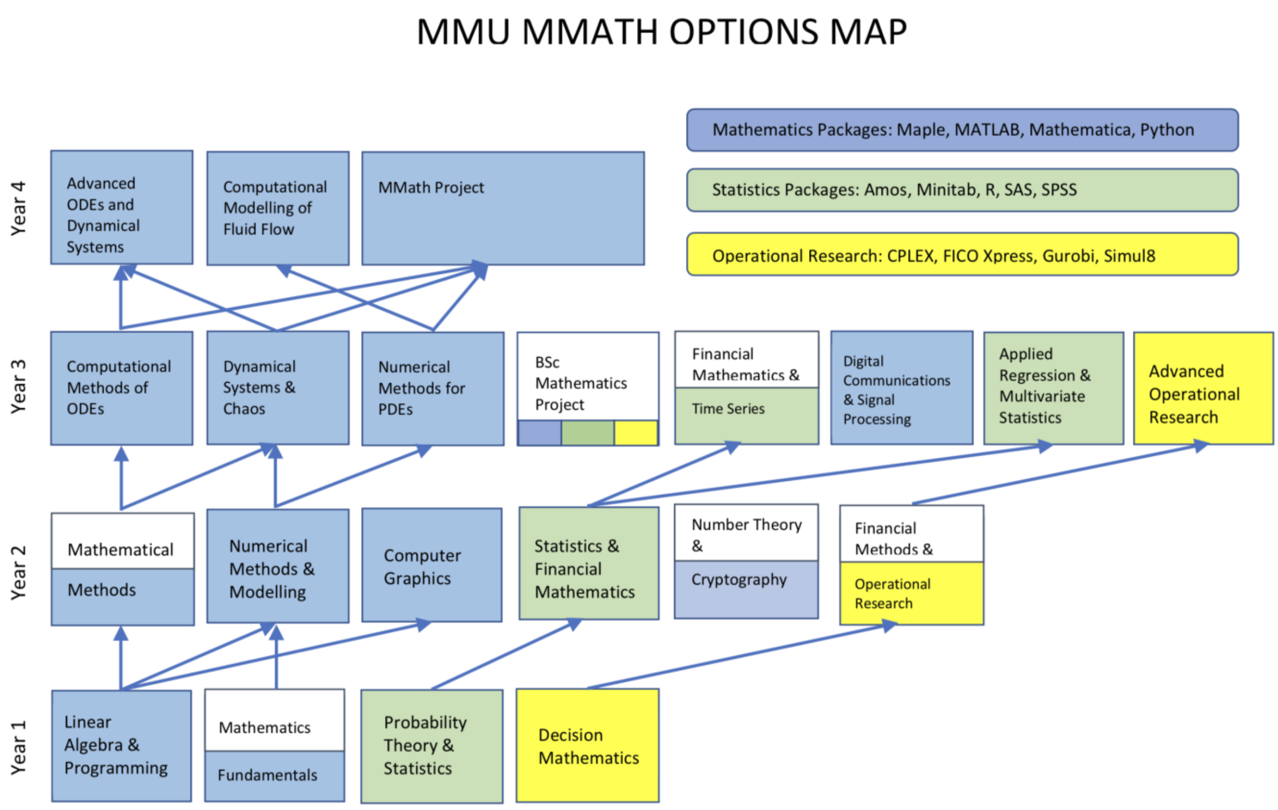

Figure 1. A typical curriculum map for Mathematics modules at MMU indicating that programming and computational modelling underpin the whole curriculum. Each module (block) is worth 30 credits. The emphasis is on programming for Mathematics and not programming for Computer Science.

The key unit in the curriculum map in Figure 1 is the Linear Algebra and Programming unit in Year 1. The students are first shown how to use MATLAB/Python as a graphing calculator and in every Linear Algebra class, MATLAB/Python functions are used to solve simple problems and plot figures. Within a few weeks, students are introduced to three programming constructs: (i) defining functions; (ii) for and while loops; and (iii) if, then, elif, else statements. The students are assessed using two in-class tests and one summer examination. One of the in-class tests set in a laboratory is based on simple programming ensuring that the students understand the three basic programming constructs (i) – (iii) above. Towards the end of the unit, programming is used in Linear Algebra to demonstrate linear transformations, in particular, rotation and translation of objects in two-dimensional space. This provides a precursor to the Year 2 unit entitled Computer Graphics – and students are given an insight into what is studied there. MATLAB/Python is also used as a graphing calculator in the other Year 1 units where specialist software in statistics and operational research is also introduced for computational modelling.

In Year 2, MATLAB/Python is used in all six units at various levels and the students are shown how programming and computational modelling can be used throughout the mathematics curriculum.

It is in Years 3 and 4 that the students start to see the real benefits of programming and computational modelling. The units Computational Methods of ODEs, Dynamical Systems and Chaos, Numerical Methods for PDEs and Digital Communications and Sound Processing are heavily MATLAB/Python-based. The coursework for these Year 3 units require some extensive programming and interpretation of solutions to real-world problems and the examinations take place in a computer laboratory with access to MATLAB/Python. Statistics and operational research packages are used for computational modelling in the units Financial Mathematics and Time Series, Applied Regression and Multivariate Statistics and Advanced Operational Research but there is far less programming in these units. This allows students a route through the curriculum if they are averse to programming. There is an extensive list of mathematics projects offered in Year 3 – some are heavily programming/computational modelling based whilst others avoid programming and modelling all together. In Year 4, the units Advanced ODEs and Dynamical Systems, Computational Modelling of Fluid Flow and the MMath project are all heavily programming/computational modelling based and the students are expected to reproduce results from recently published journal papers. In the first two units, the examinations again take place in a computer laboratory. In the next section, we will show exemplars of coursework and examination questions used at MMU.

3. Learning and Teaching and Assessment

A typical 30 credit unit will consist of one/two hours of lectures followed by two/one hours in a computer laboratory. Lectures can have between 30-120 students present and the computer laboratories have a typical capacity for 30 students - this means that some laboratory sessions have to be repeated. Staff have found that teaching in a computer laboratory is the best medium to get to know the students more intimately – better even than one-to-one tutorials. The students happily talk and help one another and there is generally a very relaxed atmosphere enabling closer interactions. Students seem very happy with this method of teaching, since the inception of the Student Union Teaching Awards in 2011, mathematics has been shortlisted on six separate occasions winning the award in 2012 and 2018. In 2015, Mathematics at MMU was visited by a Teaching Fellow form University College London (UCL) and an IEEE conference paper was published in 2016 comparing the incorporation of programming into curricula at both MMU and UCL (Nyamapfene, 2016). Some exemplars of examination and coursework questions involving programming and computational modelling will now be listed for illustration.

As Linear Algebra and Programming Skills are so vital to the curriculum, we start with an examination question from that Year-1 unit.

Examination Question 1:

(a) A geometric progression has first term *a* and common ratio $r < 1$, and the sum of the first *n* terms is given by the expression:

$$S_n = \frac{a(1-r^n)}{1-r}.$$

Write a MATLAB/Python program which can input the first term, a, and compute and output the sum of the values of r between $r = 0.1$ and $r = 0.7$ in steps of 0.2, and values of n between $n = 5$ and $n = 20$, in steps of 5. The results should appear in three columns for r, n and S_n, with suitable column headings. The value of r should be printed to one decimal place and the sum, S_n, to five decimal places. The program should use two nested **for loops**. Save the file as **exam1a.m** or **exam1a.py**, and run the program with $a = 4$, and output the results to the file **exam1a.txt**.

[Program 8 marks, Result 2 marks]

(b) Modify your part (a) program such that values of a and r are inputted from the keyboard and use a **while loop** to compute the sum, S_n, for a range on n values, starting at $n = 3$, and increasing in steps of 3. The loop should continue while the absolute difference between the part (a) sum and the sum to infinity, $S_\infty = \frac{a}{1-r}$, is greater than 0.5×10^{-6}. The table now only needs to show the values of n and S_n, again shown to five decimal places. Save the program as **exam1b.m** or **exam1b.py**, and run the program with $a = 10$ and $r = 0.5$, and output the results to the file **exam1b.txt**.

[Program 8 marks, Result 2 marks]

The next question is taken from a Year-2 examination for Operational Research and Financial Mathematics.

Examination Question 2:

The local government operational research team are developing a Markov Chain model to describe the way that the city residential population is changing. It is predicted that on average 5% of suburbanites will want to move into the centre each year and 3% of centre dwellers will want to move to the suburbs. Currently, 30% of the population live in the centre and 70% live in the suburbs.

(a) (i) Draw a Markov Chain model for the problem and write down the single stage transition matrix for it. (ii) Write a MATLAB/Python program that determines the number of years required for the proportions of centre dwellers and suburbanites to become stable and state the stable proportions. Work to four decimal places and save your file as **exam2a.m** or **exam2a.py**.

[10]

(b) A flat screen TV manufacturer offers a complete replacement warranty if a TV fails within 2 years. Based on compiled data the company has noted that 4% of the TVs fail during the first year. Whereas, 1% of TVs that survive the first year will fail during the second year. The warranty does not cover replacement TVs. (i) Draw a network representation of the problem. Hint you should have four states. (ii) Derive the fundamental matrix *N* and matrix *B*. (iii) From the matrix *B,* give an estimate of the number of TVs that will require replacing.

[10]

The next question is a coursework question from the Year-3 unit Applied Regression and Multivariate Analysis.

Coursework Question 1:

Suppose that a random variable *Y* satisfies $Y \sim Bin(3, p)$. In this case,

$$E[Y] = 3p, Var[Y] = 3p(1-p).$$

In contrast to the above, a quasibinomial model can be constructed that satisfies

$$E[Y] = 3p, Var[Y] = 3p(1-p)\phi.$$

(a) For what values of ϕ is this distribution: (i) Over-dispersed? (ii) Under-dispersed?

(b) Show that $R = 1 + Y$, satisfies

$$E[R] = 1 + 3p, Var[Y] = 3p(1-p)\phi,$$

where the mean and variance of Y can be derived from part (a). Table 1 relates the Research Excellence Framework ratings of submitted journal papers against the journal ranking using the Association of Business Schools journal list.

(c) Enter this data into R using the labels ABS and REF, and include a screenshot of your R workspace once you have done this.

Table 1: Research Excellence Framework Data

REF Rating	ABS 4*	ABS 3*	ABS 2*	ABS 1*	Unrated
4*	94	80	4	2	3
3*	95	296	29	1	6
2*	47	150	54	9	37
1*	3	28	10	6	21

(d) Using the command **sample(1:975, 1)**, delete one of the datapoints at random from each of the series ABS and REF. Include the details in your coursework submission.

(e) Using a logistic quasibinomial model test for a relationship between y and the ABS journal rating, determine the effect of including a "-1"-term in the formula for the explanatory variables on the right-hand side of the equation.

(f) Using the model output in part (e), copy and complete the following table:

ABS Rating	E[R]	Var[R]
4*		
3*		
2*		
1*		
Unrated		

[TOTAL=25 Marks]

The final examplar in this section presents a coursework question for the Year-4 unit, Advanced Ordinary Differential Equations and Dynamical Systems.

Coursework Question 2:

Consider the following nonlinear Leslie population model, where the decay rates decay exponentially with population size and the population is observed once per year:

$$X^{(k+1)} = \begin{pmatrix} b_1 e^{-rN} & b_2 e^{-rN} & b_3 e^{-rN} \\ c_1 & 0 & 0 \\ 0 & c_2 & 0 \end{pmatrix} X^{(k)},$$

where $X = (x, y, z)^T \in R^3$, is a vector of the k'th age class populations.

(a) Given that $b_1 = 5, b_2 = 50, c_1 = 0.8, c_2 = 0.6, r = 0.1, N = x + y + z,$ determine the fixed points of period one and their stability.

(b) Using the same parameters as in (a), given that, for the scaled population $x^{(0)} = 5, y^{(0)} = 15,$ and $z^{(0)} = 5$: (i) compute the number of females in each age class after 100 years; (ii) plot a graph showing how the populations evolve and give a physical interpretation.

(c) Given that $b_1 = 50, b_2 = 50, c_1 = 0.8, c_2 = 0.6, r = 0.1, N = x + y + z,$ (i) use MATLAB/Python to produce an animation in 3-dimensional space which shows how the phase portrait changes as the parameter $b_2$increases from $b_2 = 50$ to $b_2 = 80$. Initially take, $x^{(0)} = 5, y^{(0)} = 15,$ and $z^{(0)} = 5$; (ii) take still shots of the animation to submit your coursework and describe what happens to the age class populations physically.

[TOTAL=25 Marks]

On reflection, teaching in lectures, tutorials and computer laboratories provides a great deal of variety for both our students and staff and we believe that this variety is a major factor in what makes the mathematics degree loved by our students. The assessments can be made more challenging and can be applied to real-world problems, which the students appreciate. Finally, the students and parents who come to us for Open Days and Visit Days tell us that the highlight of the day is the interactive Python sessions in our laboratories.

4. Conclusion and Future Enterprises

The course philosophy for mathematics at MMU is to provide an applications-based approach using computational modelling and programming to solve real-world problems. The curriculum map displayed in Figure 1 and discussed in Section 2 demonstrates how this is possible. Section 3 presents exemplars of coursework and examination questions for a number of units from Years 1 to 4. Interested readers will find my Python book (Lynch, 2018) useful for material for Dynamical Systems units in Years 3 and 4. The book also includes a chapter listing 20 coursework exemplars as well as three example examination papers. From 2020, we will be offering an annual summer term workshop on Python programming for other departments in the Faculty of Science and Engineering at MMU. Working with Professor Louise Walker at the University of Manchester, we will also be offering annual summer workshops on "*Python for A-Level Mathematics and Beyond,*" suitable for both teachers and sixth form students. The goal is to get more students to choose mathematics at degree level.

5. Acknowledgements

This paper is based on a talk given at the ‘Programming in the Undergraduate Mathematics Curriculum’ workshop, Middlesex University, 27th June 2019.

6. References

Balon, B. and Simic, M., 2019. Using Raspberry Pi computers in education. *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics*, pp.671-676. https://doi.org/10.23919/MIPRO.2019.8756967.

Bond, P. ed., 2018. *The Era of Mathematics – Review Findings on Knowledge Exchange in the Mathematical Sciences*. Engineering and Physical Sciences Research Council and the Knowledge Transfer Network. Available at: https://admin.ktn-uk.co.uk/app/uploads/2018/04/KE-booklet-for-web.pdf [Accessed 25 March 2020].

Buteau, C., Muller, E. and Ralph, B., 2015. Integration of programming in the undergraduate Mathematics program at Brock University. *Online Proceedings of the Maths + Coding Symposium*. London (Canada). Available at: http://researchideas.ca/coding/docs/ButeauMullerRalph-Coding+MathProceedings-FINAL.pdf [Accessed 25 March 2020].

Cline, K., Fasteen, J., Francis, A., Sullivan, E. and Wendt, T., 2019. Integrating programming across the undergraduate mathematics curriculum. *PRIMUS: Problems, Resources, and Issues in Mathematics Undergraduate Studies*, 30(7), pp.735-749. https://doi.org/10.1080/10511970.2019.1616637.

Jones, L.B. and Hopkins, B.J., 2019. Teaching a course in mathematical programming. *PRIMUS: Problems, Resources, and Issues in Mathematics Undergraduate Studies*. Available at https://doi.org/10.1080/10511970.2019.1619207.

Lynch, S., 2018. *Dynamical Systems with Applications using Python*. New York: Springer International Publishing.

Nyamapfene, A. and Lynch, S., 2016. Systematic integration of MATLAB into undergraduate mathematics teaching: Preliminary lessons from two UK institutions. *IEEE Educon 2016*, pp.1145-1148. https://doi.org/10.1109/EDUCON.2016.7474699.

Peplow, M. ed., 2018. *Computational Modelling: Technological Futures*. Government Office for Science and Council for Science and Technology. Available at https://www.gov.uk/government/publications/computational-modelling-blackett-review [Accessed 25 March 2020].

Sangwin, C.J. and O’Toole, C., 2017. Computer programming in the UK mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, 48(8), pp.1133-1152. https://doi.org/10.1080/0020739X.2017.1315186.

Scratch.mit.edu, 2019. Community statistics at a glance. Available at: https://scratch.mit.edu/statistics/ [Accessed 25 March 2020].